Accredited

AS and A LEVEL

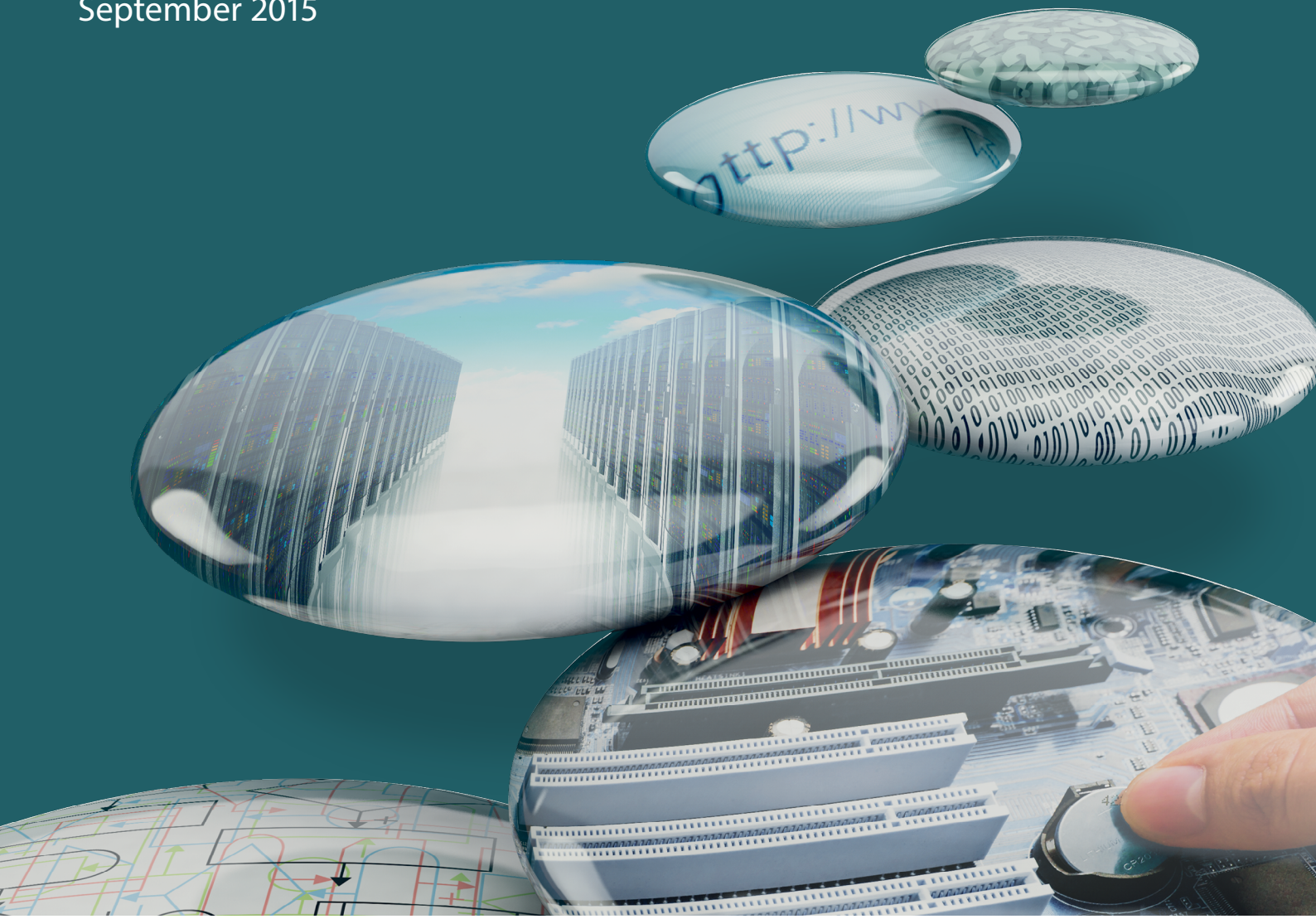
Topic Exploration Pack

H046/H446

COMPUTER SCIENCE

Theme: Structure and Function of the Processor

September 2015



CAMBRIDGE UNIVERSITY PRESS
www.cambridge.org

OCR
Oxford Cambridge and RSA

We will inform centres about any changes to the specification. We will also publish changes on our website. The latest version of our specification will always be the one on our website (www.ocr.org.uk) and this may differ from printed versions.

Oxford Cambridge and RSA Examinations is a Company Limited by Guarantee.
Registered in England. Registered company number 3484466.

Registered office: 1 Hills Road
Cambridge
CB1 2EU

OCR is an exempt charity.

Contents

This Topic Exploration Pack should accompany the OCR resource ‘Structure and Function of the Processor’ learner activities, which you can download from the OCR website.

This activity offers an opportunity for English skills development.

Structure and Function of the Processor

Learners need to understand the key components of a typical Von Neumann processor. These should include the Arithmetic Logic Unit (ALU), Control Unit (CU) and registers: the Accumulator (ACC), Program Counter (PC), Memory Data Register (MDR), Memory Address Register (MAR) and Current Instruction Register (CIR). It is important to view components as interconnected components that function together to execute programs. They should also appreciate the importance of RAM in storing running programs.

Alternative processor architectures such as Harvard should provide a deeper understanding of how programs are run and some of the potential constraints on performance. Learners should also be made aware of energy consumption considerations and problems caused by excess waste heat. In particular, the Von Neumann bottleneck should be discussed and the potential problems associated with Harvard processors separating data and instructions in a program.

The Fetch Decode Execute cycle should be explained in terms of the CPU components. Kinaesthetic activities are a good way of reinforcing understanding of the process. The cyclic nature of the process and timing of events allows the learners to appreciate the importance of clock speed in processor performance. However, as they will realise, this is not the only measure and that caching recently used instructions will also decrease fetch time particularly in programs containing iterations.
Learners should have an opportunity to use the Little Man Computer Simulator, which can be viewed at:
http://www.yorku.ca/sychen/research/LMC/LittleMan.html

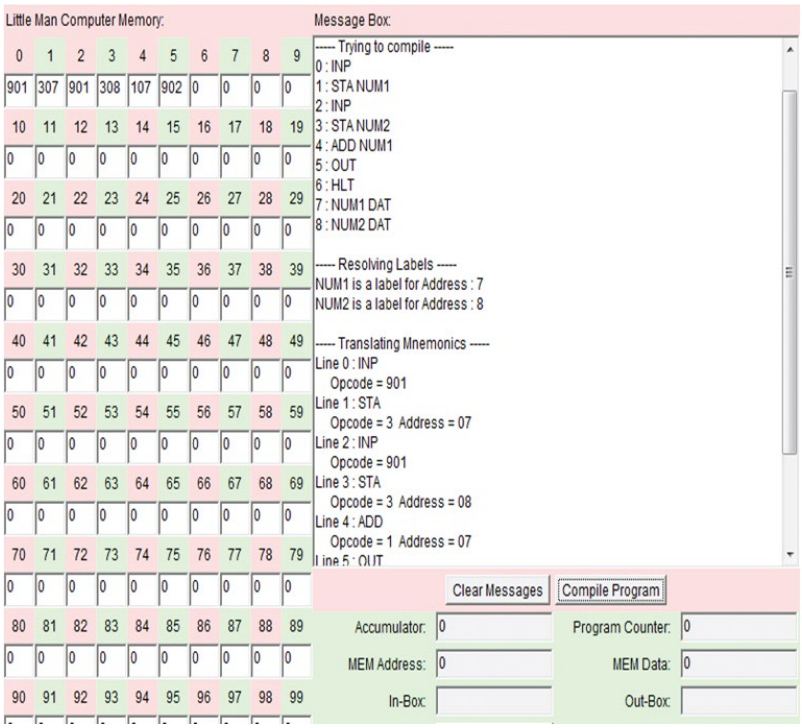


Fig 1 Little Man Computer Simulator

This will enable them to appreciate how programs written in a low-level language are executed. They will understand how instructions are broken down into the mnemonic which is translated to an opcode, and the associated memory address. One benefit of using the simulator is that learners can see all components, their contents and the program output.

Instructions executed sequentially may lead to inefficiency and underused resources. Pipelining can reduce these problems by optimising resources and use CPU time more efficiently. Instruction Level Parallelism and Thread Level Parallelism should both be covered. Modern CPU architecture includes the use of multiple cores and embedded systems.

Suggested Activities

Activity	Learning Objective	Resources Required	Additional Activities and Links
CPU architecture interactive poster	Learners will know about the structure of the processor.	Coloured paper and other stationery	The end product could also be an interactive presentation, animation or authored multimedia product. It could also be developed into a mobile app using software such as MIT App Inventor.
CPU component name game	To reinforce key component names associated with CPU architecture	Small adhesive note pages	This activity could be used as a revision activity.
Fetch Decode Execute class simulation	Learners will understand the Fetch Decode Execute cycle.	Cards with components of the processor and program instructions and data	Learners all assume the role of a CPU sub-component. A simple low-level program is then executed on this 'human virtual processor'.

Activity	Learning Objective	Resources Required	Additional Activities and Links
Understanding the Little Man Computer Simulator	Learners will understand how the processor runs programs.	Internet connected computer	Complete a series of programming exercises to simulate a variety of different low level programs including those with selection and iteration.
Benchmark testing a CPU	Learners will know how to benchmark test a CPU.	Free download CPU performance test software.	Software can be downloaded from http://novabench.com/ or http://www.sisoftware.co.uk/ (Lite evaluation copy). Other products are available but may be 30-day evaluation copies.
Pipelining demonstration	Learners will understand how pipelining improves efficiency.	Paper, scissors, pens, adhesive putty	Additional information can be found at: http://www.gamedev.net/page/resources/_/technical/general-programming/a-journey-through-the-cpu-pipeline-r3115 http://www.cs.iastate.edu/~prabhu/Tutorial/PIPELINE/pipetitle.html
Alternative processor architectures	Learners will be able to appreciate the advantages and limitations associated with different processor architectures.	Internet connected computers	Research and compare different processor architectures and produce a simple guide in the form of a magazine article.

Approaches to Delivery

This topic can be delivered through the use of simulators and activities that can present simplified illustrations of complex structures and actions. This is of benefit and can aid understanding although some of the detail is lost through over-simplification. For example, in the Fetch Decode Execute simulation, a single core processor with no cache is used.

If Raspberry Pi computers are available, these can be overclocked using preset parameters. However, care should be taken that these are set correctly. It is also not advisable to leave processors in an overclocked state permanently without additional cooling as it may reduce lifespan. A simple program can be used to benchmark test the processor.

Pipelining is an important concept in modern processors. Learners will appreciate the limitation of the Von Neumann architecture and that parallelism permits faster processing. Any simple activity could be used to illustrate the basic principles. In the activity a production line is used to make folded paper helicopters. The activity will show that pipelining permits more ‘processes’ to be executed per unit time.

Activity 1: CPU architecture interactive poster

Objectives

Learners will know the structure of a typical processor and be able to name and give the function of key components

Resources

- display paper
- pins
- thin string

Main Activity

Learners use resources to produce an interactive poster showing the structure of a typical processor. They should include:

- ALU
- Control Unit
- Program Counter
- Accumulator
- Memory Data Register
- Memory Address Register
- Current Instruction Register

Each component should have its function associated with it. This could be added by placing a “flap” of paper over a component that is lifted to show its function.

As an alternative activity, an interactive presentation or mobile phone app could be produced. In both cases, the image of the component should be clickable to create a pop-up showing the function.

Key questions

- What are the advantages of using registers to hold specific items within the CPU?
- What is the minimum size that a register can be?
- What is the purpose of the Program Counter and why is it needed?
- Give an example of a logical operation?
-

Activity 2: Benchmark testing a CPU

Objectives

Learners will know how to measure the performance of a typical CPU.

Resources

- Windows PC
- Downloaded benchmark test software from: http://novabench.com/ or http://www.sisoftware.co.uk/
- Raspberry Pi computer

Main activity

Learners will use a downloaded program to test the performance of a Windows PC. They will then compare the performance with a Raspberry Pi benchmark test. They will be introduced to some standard measurements of CPU performance: MIPS (millions of instructions per second) and GFLOPS (giga floating point operations per second) as a way of comparing performance data. They will also appreciate that some programs are more demanding to run than others and that performance tests should “stress” the processor to assess its capabilities.

Key questions

- What are the types of programs that will cause the most “stress” on the processor?
- Thinking about the Fetch Decode Execute cycle, what could be done to increase performance?
- What might cause some clock cycles to not result in the execution of an instruction?

Activity 3: Illustration of pipelining

Objectives

Learners will understand how pipelining improves the efficiency of a processor.

Resources

- paper
- pencil
- scissors
- ruler
- sticky putty
- stopwatch or means of recording elapsed time

Main activity

Organise learners into small teams. Each team member has a specific role. Initially, to simulate a non-pipelined execution, each step must be performed sequentially until a “helicopter” is complete. Then, the entire process can begin again. Learners should time how long it takes to make three complete “helicopters”. With pipelining, when one team member has completed his/her part in the process, the production moves on to the next step. However, the early stage can then begin again for another “helicopter”. Learners again time how long it takes to make three “helicopters”.

Key questions

- How does the use of pipelining increase the efficiency of processors?
- What is a thread?
- What is the difference between Instruction **Level Pipelining** and **Thread Level Pipelining**?

OCR customer contact centre
General qualifications
Telephone 01223 553998
Facsimile 01223 552627
Email general.qualifications@ocr.org.uk